

Proj2

ggMonet

March 3, 2016

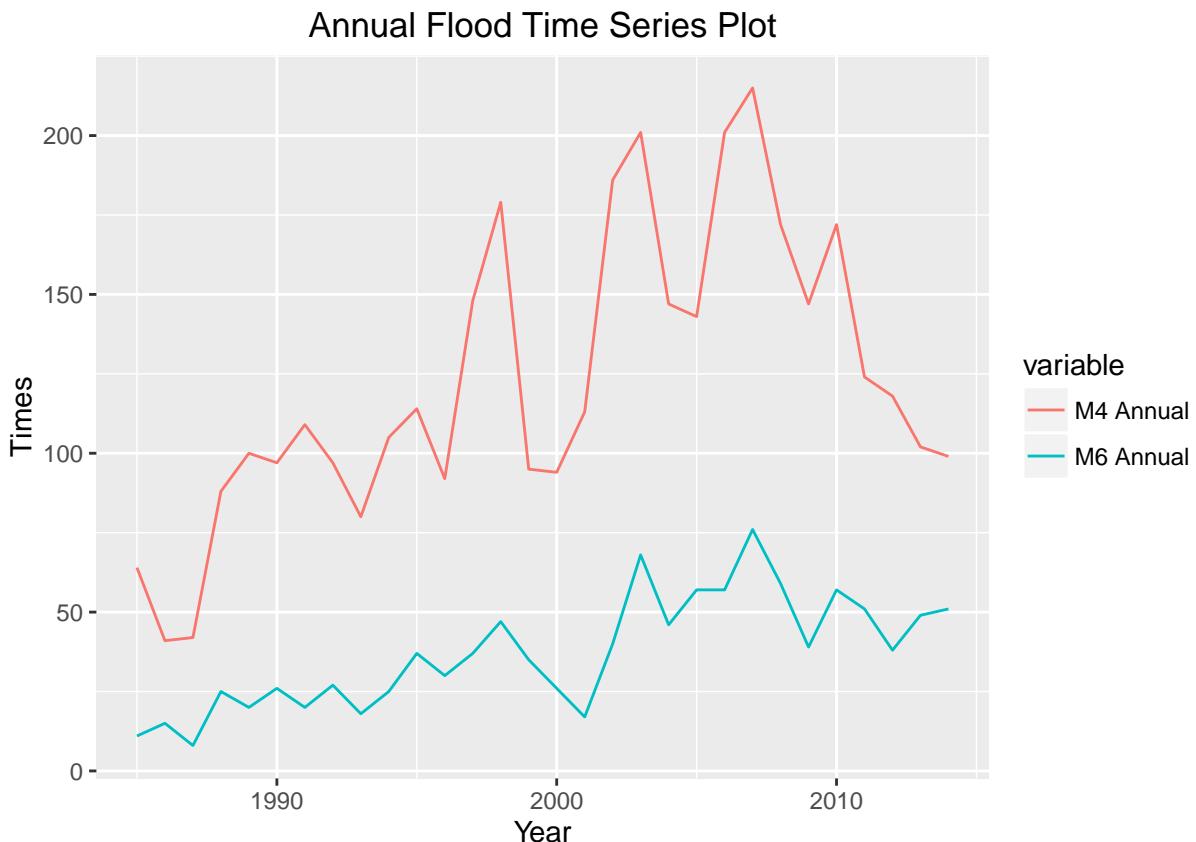
```
#####
#      Global Setup      #
#####

setwd("C:/Columbia Courses/Visualization/Project2")

#####
#  Plots about Flood Stats  #
#####

stat = read.csv("GlobalFloodsRecordAnalyses.csv", as.is = TRUE)

# Time series plot of annual floods --Xuyan
library(ggplot2)
library(reshape2)
names(stat) = c("Year", "M4 Cumulative", "M6 Cumulative", "M4 Annual", "M6 Annual")
floodAnnual = melt(stat[-(2:3)], id.vars = "Year", value.name = "Times")
ggplot(floodAnnual, aes(Year, Times)) + geom_line(aes(color = variable)) +
  ggtitle("Annual Flood Time Series Plot") + scale_fill_brewer(palette = "Set2")
```



TODO 1, scale of the plots and some more variables, and heatmap without geographical info

all plots in ggplot style

Tian and Xiyue

```
#####
# Plots about Flood Master --Global #
#####
master = read.csv("GlobalFloodsRecordMaster.csv", as.is = TRUE)

library(fields)
library(maptools)
library(ggplot2)
library(ggmap)
library(maps)
library(plyr)
library(lattice)
library(Rmisc)
library(mapproj)
library(rgdal)

# data manipulation --Tian
df = master
df$Centroid.X <- as.numeric(df$Centroid.X)
df$Centroid.Y <- as.numeric(df$Centroid.Y)
df$Severity..<- as.numeric(df$Severity..)
class(df$Centroid.X[1])

## [1] "numeric"

df <- df[-which(is.na(df$Centroid.X)),]
XLon <- as.numeric(df$Centroid.X)
YLat <- as.numeric(df$Centroid.Y)
Z <- as.numeric(df$Severity..)
Cause <- df$Main.cause
#rev(sort(table(Cause)))[1:6]
n <- length(Cause)
for (i in 1:n){
  if (grepl('eavy',Cause[i])){Cause[i] <- replace(Cause[i], grepl('eavy',Cause[i]),1) }
  #1 stands for 'Heavy Rain'
  else if(grepl('clone',Cause[i])){Cause[i] <- replace(Cause[i], grepl('clone',Cause[i]),2)}
  #2 stands for 'Tropical Cyclone'
  else if(grepl('onsoon',Cause[i])){Cause[i] <- replace(Cause[i], grepl('onsoon',Cause[i]),3)}
  #3 stands for 'Monsoon'
  else if(grepl('orrential',Cause[i])){Cause[i] <- replace(Cause[i], grepl('orrential',Cause[i]),4)}
  #4 stands for 'Torrential Rain'
  else {Cause[i] <- replace(Cause[i],TRUE,5)}
```

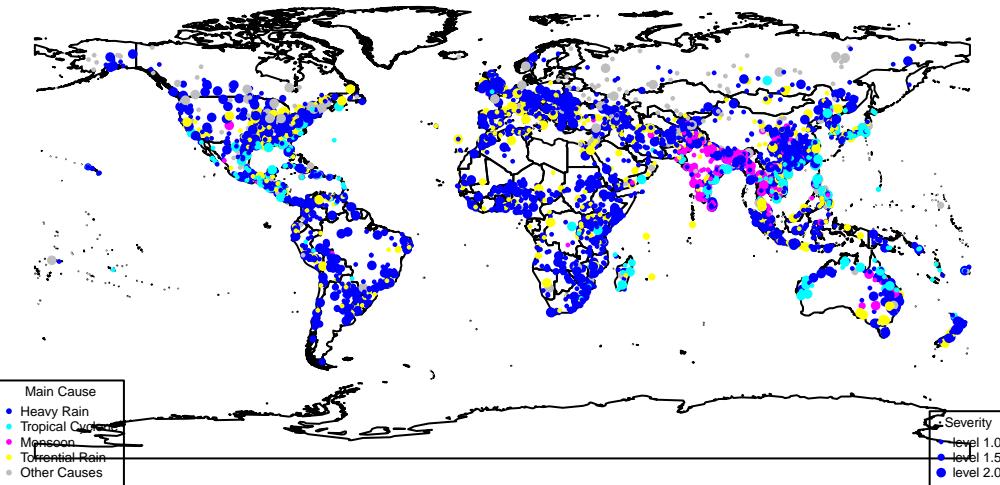
```

#5 stands for 'Other Causes'
}

#Try simple plot of "Main Causes" and "Severity": --Tian
data(wrld_simpl)
plot(wrld_simpl)
points(XLon, YLat, pch = 16, cex = Z/3, col = as.numeric(Cause)+3)
title(main = "Flood Distribution \nBased on Main Causes and Severity", cex.main =1)
legend("bottomleft",legend = c("Heavy Rain","Tropical Cyclone","Monsoon","Torrential Rain","Other Causes"),
      cex = 0.4, pch = 16, col = c(4:8), title ="Main Cause",title.adj = .5)
legend("bottomright",legend = c("level 1.0","level 1.5","level 2.0"),
      pt.cex = c(1,1.5,2)/3, cex = .4, pch = 16,col =4, title = "Severity")

```

Flood Distribution Based on Main Causes and Severity



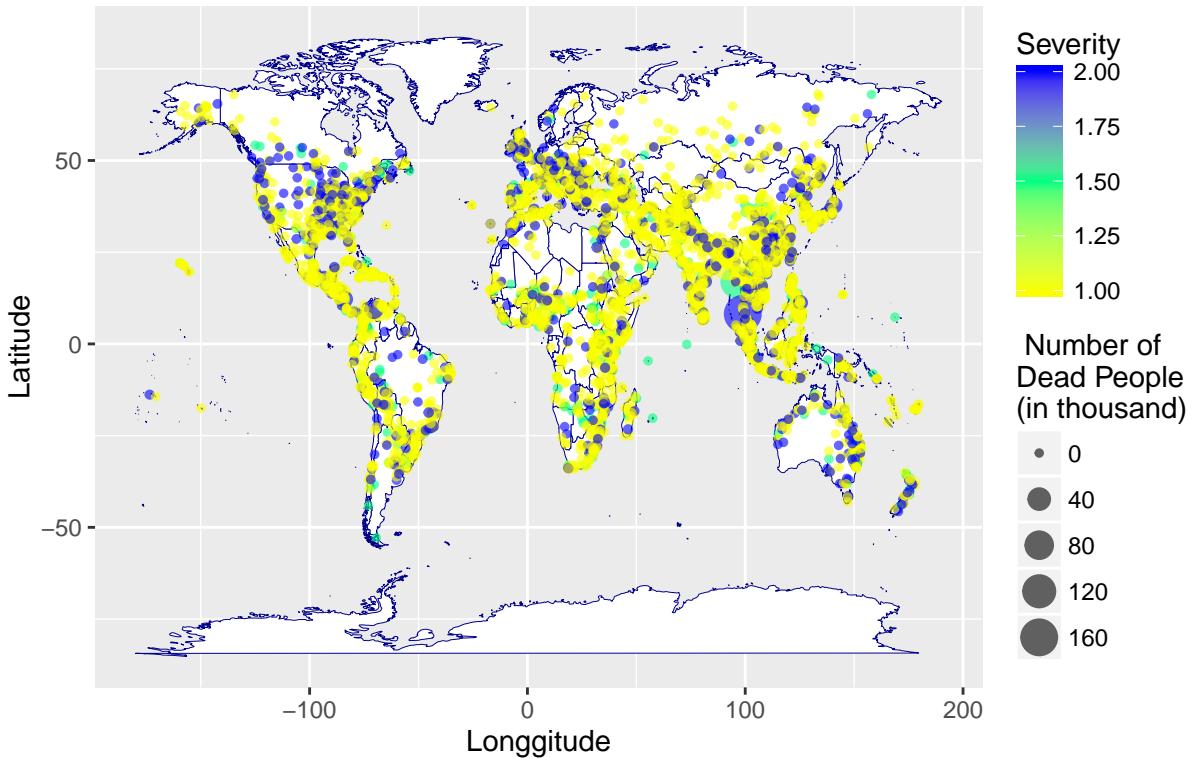
```

#Try ggplot of "Number of Dead People" and "Severity" --Tian
Dead <- as.numeric(df$Dead)
df_new <- data.frame(XLon, YLat, Z, Dead)

world <- map_data("world")
ggplot(world, aes(long, lat)) +
  geom_polygon(aes(group=group), fill = "White", color ="Dark Blue", size = 0.05) +
  geom_jitter(data=df_new, aes(XLon, YLat, color = Z, size = Dead/1000), alpha = 0.6) +
  scale_colour_gradientn(colours = rainbow(3, start = 0.17, alpha = 0.2)) +
  labs(title = "Flood Distribution with\n Number of Dead People and Severity", x = "Longgitude",
       y = "Latitude", size = " Number of\nDead People\n(in thousand)", color = "Severity")+
  theme(plot.title = element_text(lineheight=1, face="bold"))

```

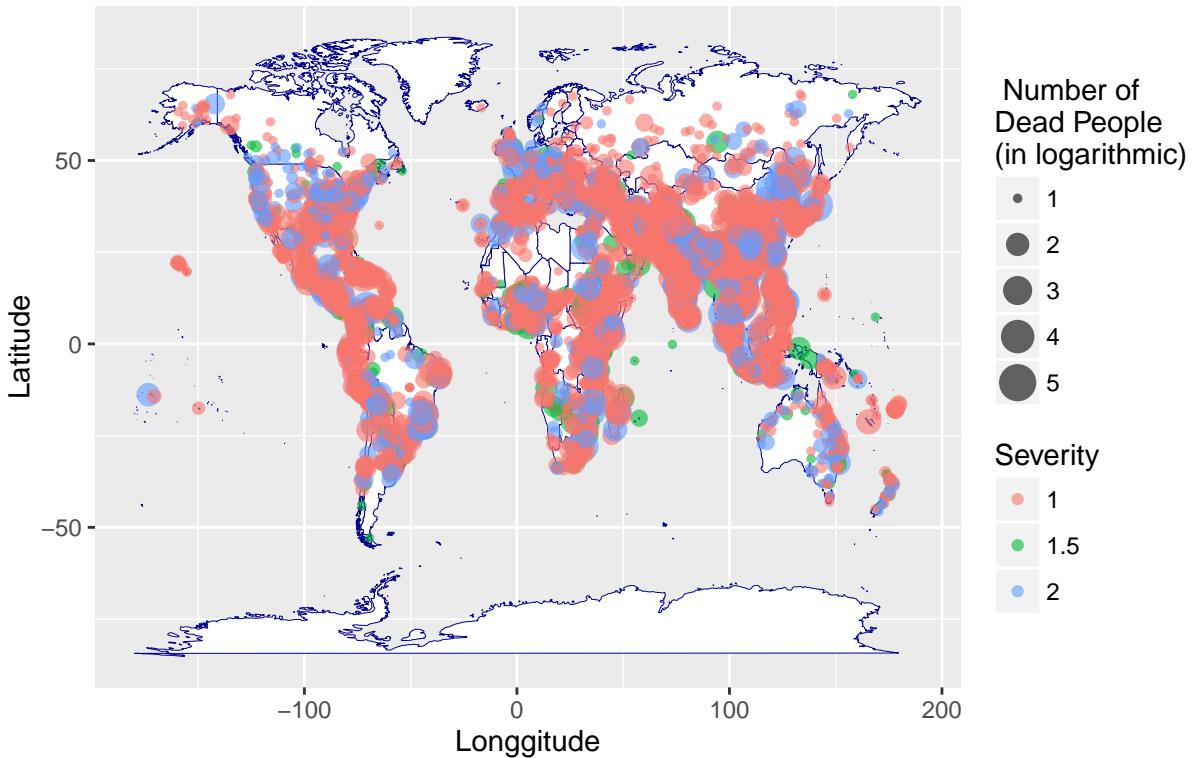
Flood Distribution with Number of Dead People and Severity



```
#Try ggplot of "Number of Dead People" and "Severity" another version --Xuyan
# severity as factor and logarithmic dead
Dead <- as.numeric(df$Dead)
df_new <- data.frame(XLon, YLat, Z, Dead)

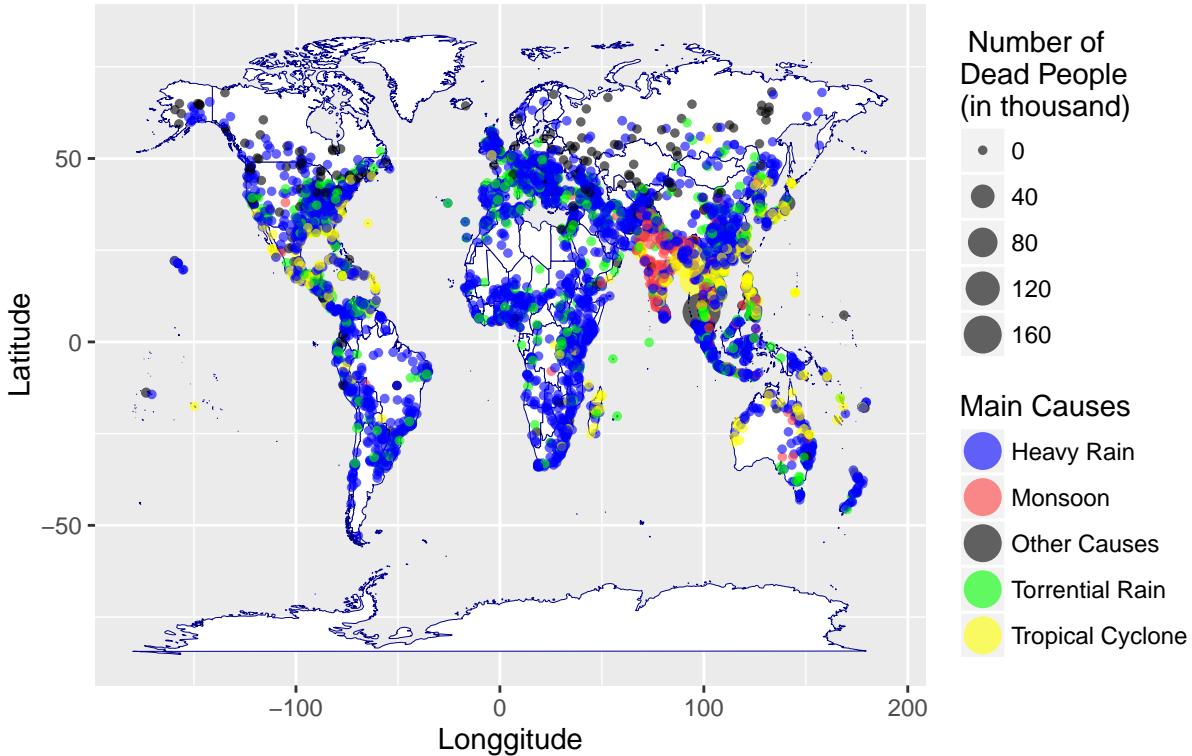
df_new$Z = as.factor(df_new$Z)
world <- map_data("world")
ggplot(world, aes(long, lat)) +
  geom_polygon(aes(group=group), fill = "White", color ="Dark Blue", size = 0.05) +
  geom_jitter(data=df_new, aes(XLon, YLat, color = Z, size = log(Dead+10,10)), alpha = 0.6) +
  # scale_colour_gradientn(colours = rainbow(3, start = 0.17, alpha = 0.2)) +
  labs(title = "Flood Distribution with\n Number of Dead People and Severity", x = "Longgitude",
       y = "Latitude", size = " Number of\nDead People\n(in logarithmic)", color = "Severity")+
  theme(plot.title = element_text(lineheight=1, face="bold"))
```

Flood Distribution with Number of Dead People and Severity



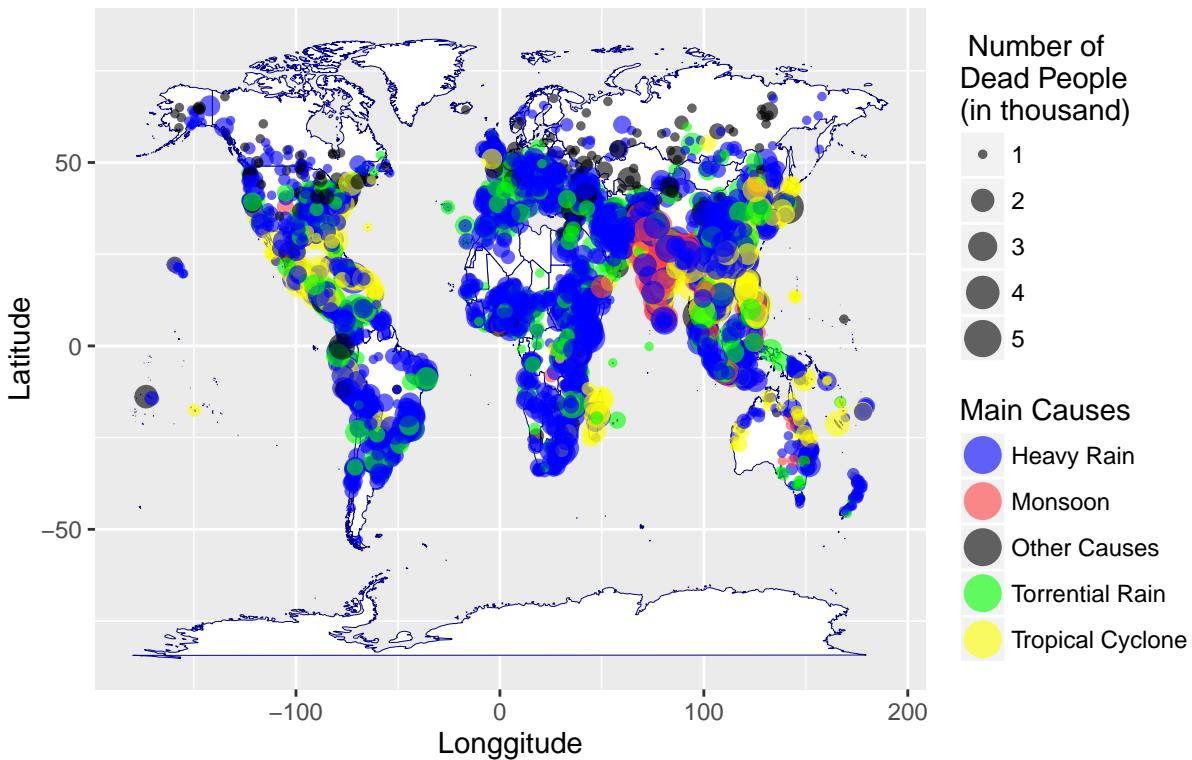
```
#Try ggplot of "Number of Dead People" and "Main Causes" --Tian
for (i in 1:n){
  Cause[i] <- replace(Cause[i], Cause[i]=='1', 'Heavy Rain')
  Cause[i] <- replace(Cause[i], Cause[i]=='2', 'Tropical Cyclone')
  Cause[i] <- replace(Cause[i], Cause[i]=='3', 'Monsoon')
  Cause[i] <- replace(Cause[i], Cause[i]=='4', 'Torrential Rain')
  Cause[i] <- replace(Cause[i], Cause[i]=='5', 'Other Causes')
}
df_new2 <- data.frame(XLon, YLat, Cause, Dead)
ggplot(world, aes(long, lat)) +
  geom_polygon(aes(group=group), fill = "White", color = "Dark Blue", size = 0.05) +
  geom_jitter(data=df_new2, aes(XLon, YLat, color = Cause, size = Dead/1000), alpha = 0.6) +
  scale_colour_manual(values = c("blue","brown1","black","green","yellow"))+
  labs(title = "Flood Distribution with\n Number of Dead People and Main Causes", x = "Longgitude",
       y = "Latitude", size = " Number of\nDead People\n(in thousand)", color = "Main Causes")+
  guides(colour = guide_legend(override.aes = list(size=6)))+
  theme(plot.title = element_text(lineheight=1, face="bold"))
```

Flood Distribution with Number of Dead People and Main Causes



```
#Try ggplot of "Number of Dead People" and "Main Causes" --Xuyan
for (i in 1:n){
  Cause[i] <- replace(Cause[i], Cause[i]=='1', 'Heavy Rain')
  Cause[i] <- replace(Cause[i], Cause[i]=='2', 'Tropical Cyclone')
  Cause[i] <- replace(Cause[i], Cause[i]=='3', 'Monsoon')
  Cause[i] <- replace(Cause[i], Cause[i]=='4', 'Torrential Rain')
  Cause[i] <- replace(Cause[i], Cause[i]=='5', 'Other Causes')
}
df_new2 <- data.frame(XLon,YLat,Cause,Dead)
ggplot(world, aes(long, lat)) +
  geom_polygon(aes(group=group), fill = "White", color = "Dark Blue", size = 0.05) +
  geom_jitter(data=df_new2, aes(XLon, YLat, color = Cause, size = log(Dead+10,10)), alpha = 0.6) +
  scale_colour_manual(values = c("blue","brown1","black","green","yellow"))+
  labs(title = "Flood Distribution with\n Number of Dead People and Main Causes", x = "Longgitude",
       y = "Latitude", size = " Number of\nDead People\n(in thousand)", color = "Main Causes")+
  guides(colour = guide_legend(override.aes = list(size=6)))+
  theme(plot.title = element_text(lineheight=1, face="bold"))
```

Flood Distribution with Number of Dead People and Main Causes



TODO 2, more plots on the distribution of countries ie density

Xuyan

```
#####
# Reason for Local Plots #
#####

# country_cleansing
library(plyr)
country = master
country$Dead = as.numeric(country$Dead)
country = country[!is.na(country$Dead),]

country$Country = gsub("[?]", "", country$Country)
country$Country = gsub("[/]", "", country$Country)
country$Country = gsub("^", "", country$Country)
country$Country = gsub(" $", "", country$Country)
country$Country[country$Country == "USA."] = "USA"

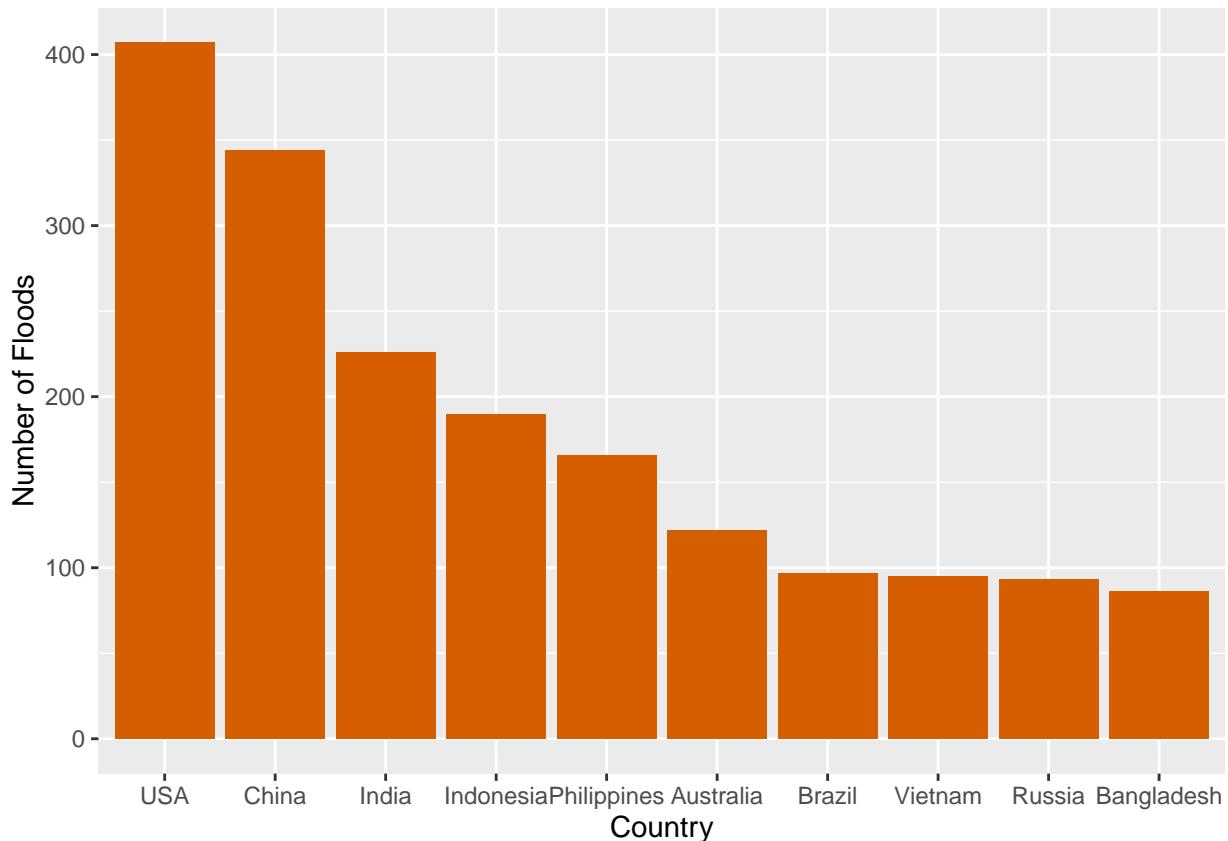
library(dplyr)
library(scales)
```

```

country_group = group_by(country, Country)
country_summary = summarize(country_group, num = n(), dead = sum(Dead))
country_summary = as.data.frame(country_summary[order(country_summary$num, decreasing=T),])

ggplot(country_summary[1:10], aes(reorder(Country, -num), num)) + geom_bar(stat = "identity", fill = "#D55E00")
  ylab("Number of Floods") + xlab("Country")

```



```

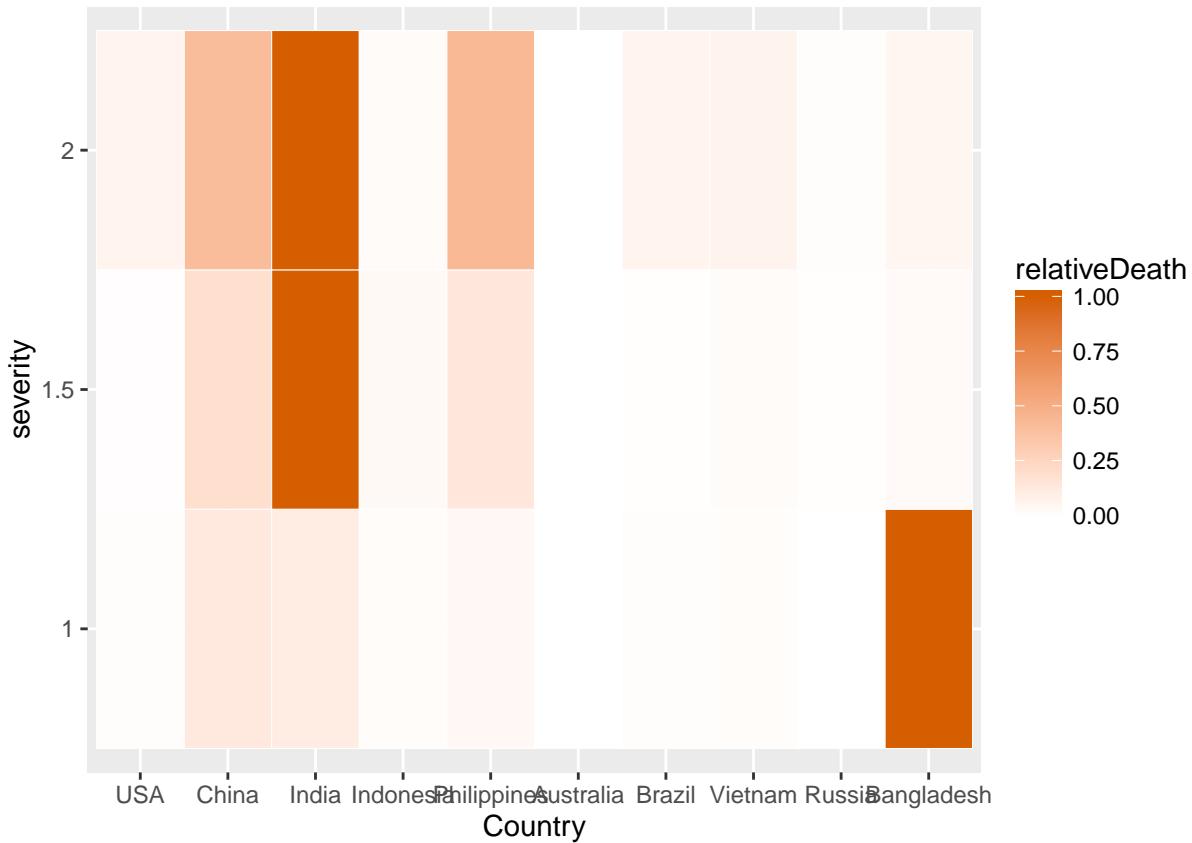
#List the top3 countries

# heatmap of country + Severity + death
country_sev = group_by(country, Country, Severity..)
country_sev_sum = summarize(country_sev, num = n(), dead = sum(Dead))
names(country_sev_sum)[2] = "severity"

top10 = arrange(country_summary, desc(num))$Country[1:10]
country_sev_top10 = country_sev_sum[country_sev_sum$Country %in% top10,]
country_sev_top10 = ddply(as.data.frame(country_sev_top10), .(severity), transform, relativeDeath = res

p = ggplot(country_sev_top10, aes(reorder(Country, -num), severity)) +
  geom_tile(aes(fill = relativeDeath), colour = "white")
p + scale_fill_gradient(low = "white", high = "#D55E00")+
  xlab("Country")

```



As we can see, USA, China and India are the top three frequently impacted countries. In the next part, we will have a more detailed look into the distributions of floods happened in the three countries. And from the heatmap of the relative death (rescaled by the most death occurred in a certain severity), we can see that USA did a great job in preventing death in floods while India and Bangladesh did not.

```
#####
# Plots about Flood Master --Local #
#####
# -- Hiro

master_f = data.frame(master)

# information of USA

## getting info of usa
usa_master = master_f[master_f$Country == "USA", ]

## getting map of usa
map_usa <- get_map(location = "usa", maptype = "satellite", zoom = 4)

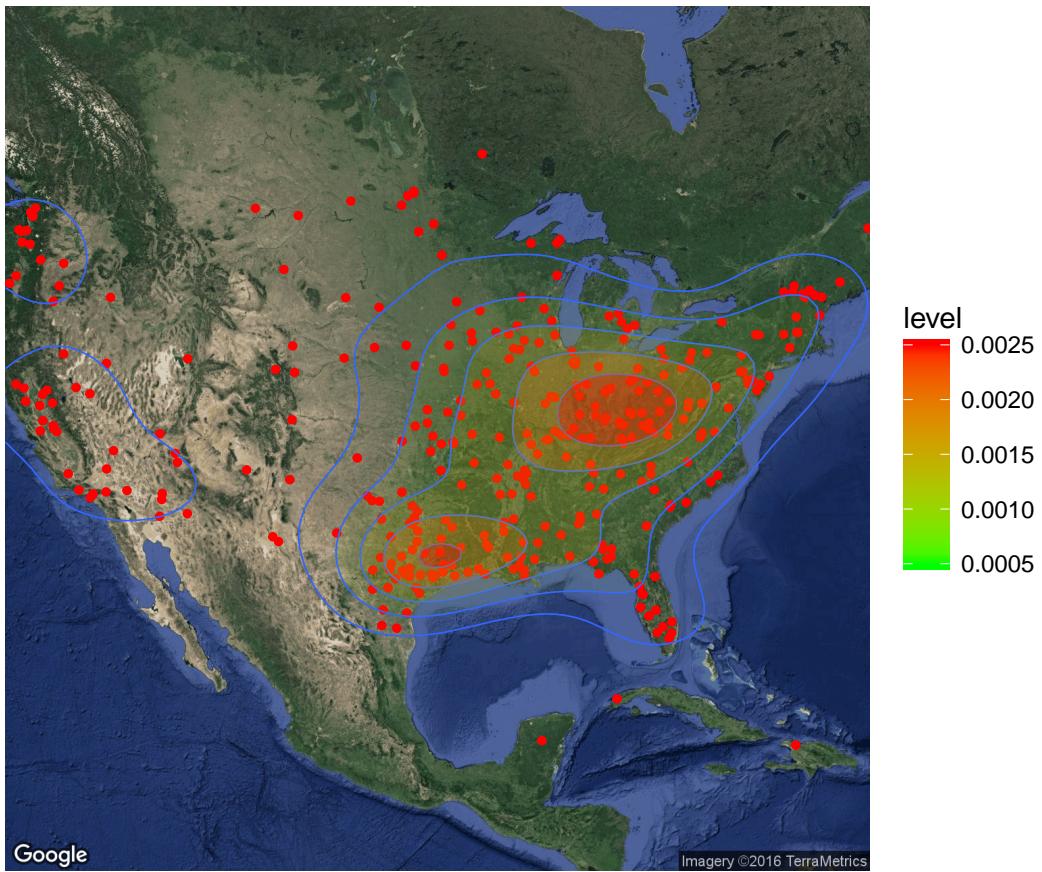
## Showing which area had floods in the past
usa_master_heat = data.frame(Centroid.X=as.numeric(usa_master$Centroid.X),
                             Centroid.Y=as.numeric(usa_master$Centroid.Y),
```

```

Total.floods.M.4=usa_master$Total.floods.M.4)

ggmap(map_usa, extent = "device") +
  geom_point(aes(x=usa_master_heat$Centroid.X,
                 y=usa_master_heat$Centroid.Y),
              data=usa_master_heat, col="red", size=1) +
  geom_density2d(data=usa_master_heat,
                 aes(x = usa_master_heat$Centroid.X,
                     y = usa_master_heat$Centroid.Y),
                 size = 0.3) +
  stat_density2d(data=usa_master_heat,
                 aes(x = usa_master_heat$Centroid.X,
                     y = usa_master_heat$Centroid.Y,
                     fill = ..level..,
                     alpha = ..level..),
                 size = 0.01, geom = "polygon") +
  scale_fill_gradient(low = "green", high = "red") +
  scale_alpha(range = c(0, 0.3), guide = FALSE)

```



```

# information of China

## getting info of china
china_master = master_f[master_f$Country == "China", ]

## getting map of china

```

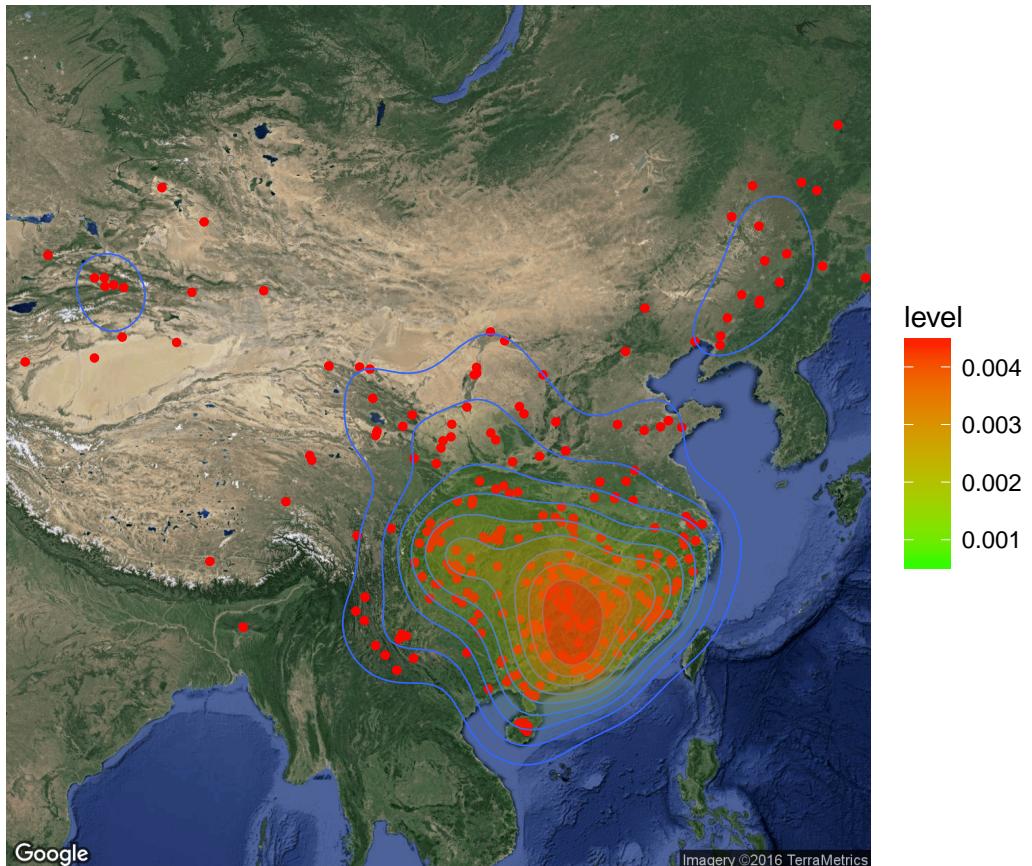
```

map_china <- get_map(location = "china", maptype = "satellite", zoom = 4)

## Showing which area had floods in the past
china_master_heat = data.frame(Centroid.X=as.numeric(china_master$Centroid.X),
                                 Centroid.Y=as.numeric(china_master$Centroid.Y),
                                 Total.floods.M.4=china_master$Total.floods.M.4)

ggmap(map_china, extent = "device") +
  geom_point(aes(x=china_master_heat$Centroid.X,
                 y=china_master_heat$Centroid.Y),
             data=china_master_heat, col="red", size=1) +
  geom_density2d(data=china_master_heat,
                 aes(x = china_master_heat$Centroid.X,
                     y = china_master_heat$Centroid.Y),
                 size = 0.3) +
  stat_density2d(data=china_master_heat,
                 aes(x = china_master_heat$Centroid.X,
                     y = china_master_heat$Centroid.Y,
                     fill = ..level..,
                     alpha = ..level..),
                 size = 0.01, geom = "polygon") +
  scale_fill_gradient(low = "green", high = "red") +
  scale_alpha(range = c(0, 0.3), guide = FALSE)

```



```

# information of India

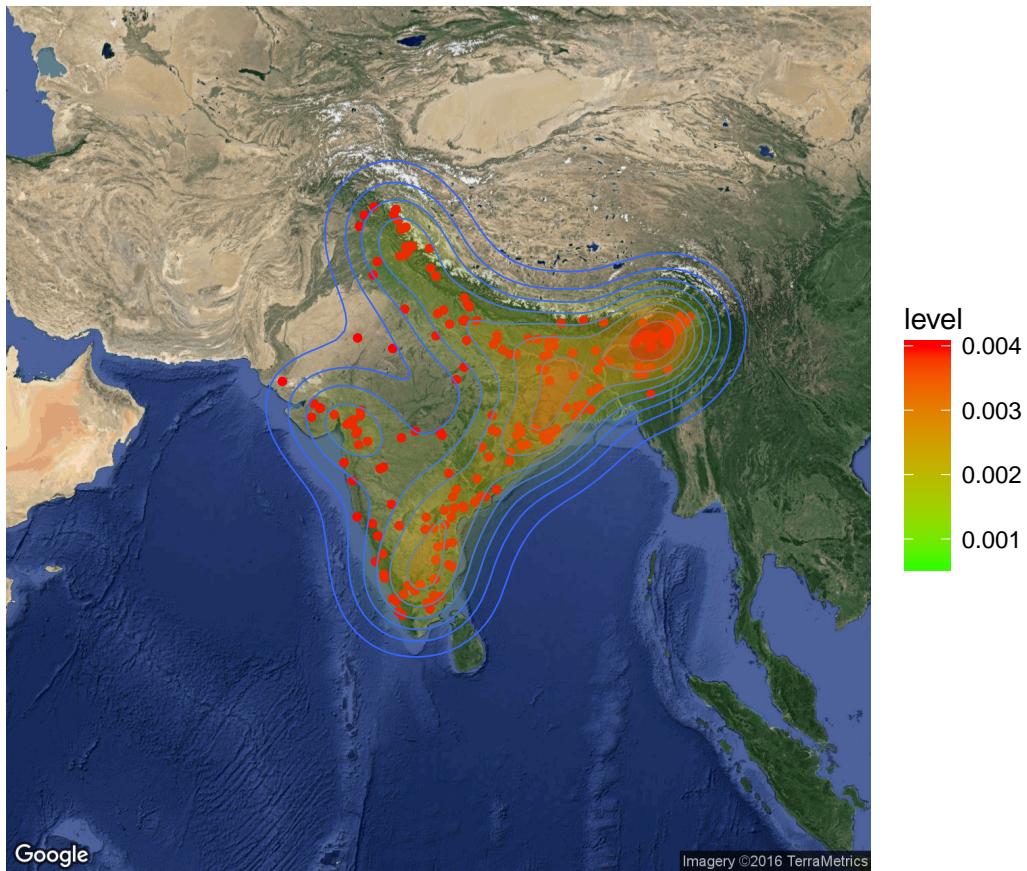
## getting info of India
india_master = master_f[master_f$Country == "India", ]

## getting map of India
map_india <- get_map(location = "India", maptype = "satellite", zoom = 4)

## Showing which area had floods in the past
india_master_heat = data.frame(Centroid.X=as.numeric(india_master$Centroid.X),
                                Centroid.Y=as.numeric(india_master$Centroid.Y),
                                Total.floods.M.4=india_master$Total.floods.M.4)

ggmap(map_india, extent = "device") +
  geom_point(aes(x=india_master_heat$Centroid.X,
                 y=india_master_heat$Centroid.Y),
             data=india_master_heat, col="red", size=1) +
  geom_density2d(data=india_master_heat,
                 aes(x = india_master_heat$Centroid.X,
                     y = india_master_heat$Centroid.Y),
                 size = 0.3) +
  stat_density2d(data=india_master_heat,
                 aes(x = india_master_heat$Centroid.X,
                     y = india_master_heat$Centroid.Y,
                     fill = ..level..,
                     alpha = ..level..),
                 size = 0.01, geom = "polygon") +
  scale_fill_gradient(low = "green", high = "red") +
  scale_alpha(range = c(0, 0.3), guide = FALSE)

```



TODO 3, plots of some days with a several occurrences and the pressure data

Xuyan

```
#####
# Plots about NOAA Data #
#####

# Read geological data

library(RNetCDF)
noaa = open.nc('NOAA_Daily_phi_500mb.nc')
data = read.nc(noaa)

xlon = data$X
ylat = rev(data$Y)
# flood data
# master = read.csv("GlobalFloodsRecordMaster.csv", as.is = TRUE)
# stat = read.csv("GlobalFloodsRecordAnalyses.csv", as.is = TRUE)

master$Began = as.Date(master$Began, format = "%d-%b-%y")
```

```

master$Ended = as.Date(master$Ended,format = "%d-%b-%y")

master_phi = master[master$Centroid.Y>min(ylat) & master$Centroid.Y<max(ylat),]

# check the distribution of begin dates
library(dplyr)
date = data.frame(date = master_phi$Began, cnt = 1)
group = group_by(date,date)
summ = summarise(group,cnt = n())
summ$date[summ$cnt == max(summ$cnt)]
```

[1] "2010-06-22" "2010-07-27"

```

# found that "1998-05-20" and "2002-06-12" is the most
maxDates = summ$date[summ$cnt == max(summ$cnt)]
```

```

# set transparency
add.alpha = function(COLORS, ALPHA){
  if(missing(ALPHA)) stop("provide a value for alpha between 0 and 1")
  RGB = col2rgb(COLORS, alpha=TRUE)
  RGB[4,] = round(RGB[4,]*ALPHA)
  NEW.COLORS = rgb(RGB[1,], RGB[2,], RGB[3,], RGB[4,], maxColorValue = 255)
  return(NEW.COLORS)
}
pal = colorRampPalette(c(rgb(0,0,1), rgb(0,1,0), rgb(1,0,0)))
COLORS = add.alpha(pal(100), 0.6)
```

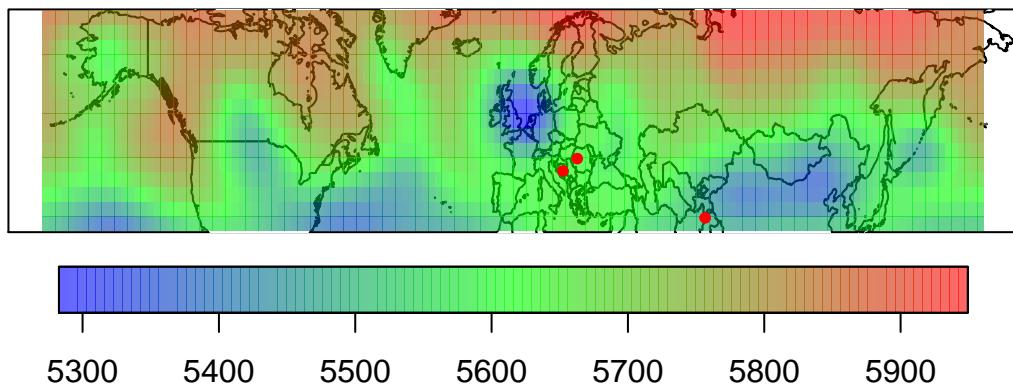
```

for(i in 1:length(maxDates)){
  maxDate = maxDates[i]
  phi3 = as.numeric(maxDate-as.Date("1948-01-01"))+1
  z = data$phi[,,phi3]

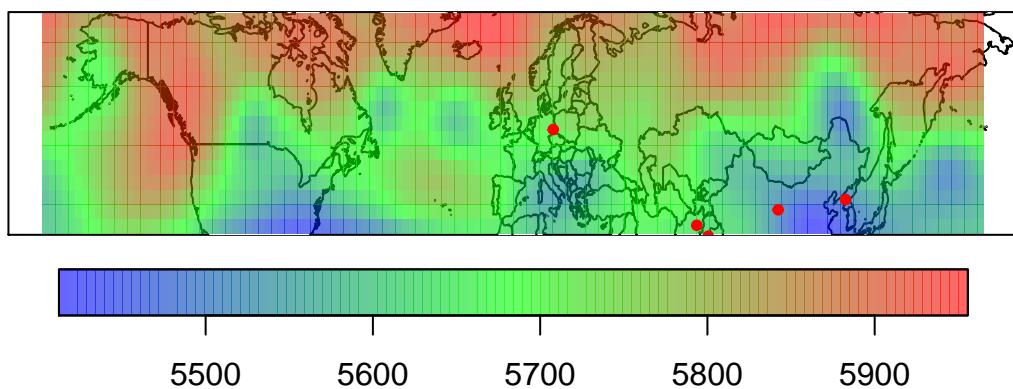
  # find the data of the floods that day
  floodDay = master[master$Began==maxDate & is.na(master$Began)==0,]

  # map data
  plot(c(min(xlon)-180,max(xlon)-180), c(min(ylat),max(ylat)), type="n", xlab="", ylab="", xaxt='n', yaxt='n', map=add=TRUE, fill=TRUE, col="white")
  image.plot(xlon-180, ylat, z, add=TRUE, col = COLORS, horizontal = T, legend.mar = 3)
  points(floodDay$Centroid.X, floodDay$Centroid.Y, pch = 20, col = "red")
}
```

2010-06-22



2010-07-27



According to the plots, we can see that many floods occurred in the areas with lower pressure, detailed analysis will be illustrated below.

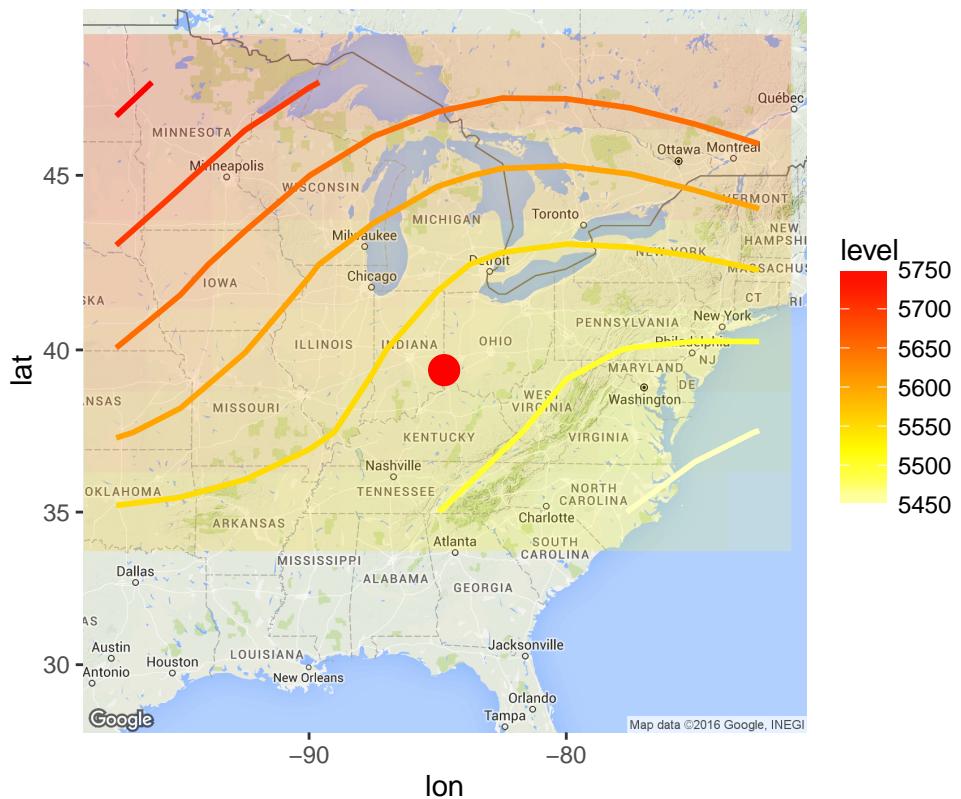
TODO 4, Countours of a certain flood within the NOAA data area

Hiro, Phoebe

Firstly, we will focus on changes of contours during the flood that happened in the area of Southern Michigan, central Indiana, and western Ohio in the US from Jun 27, 2015 to Jun 29, 2015.

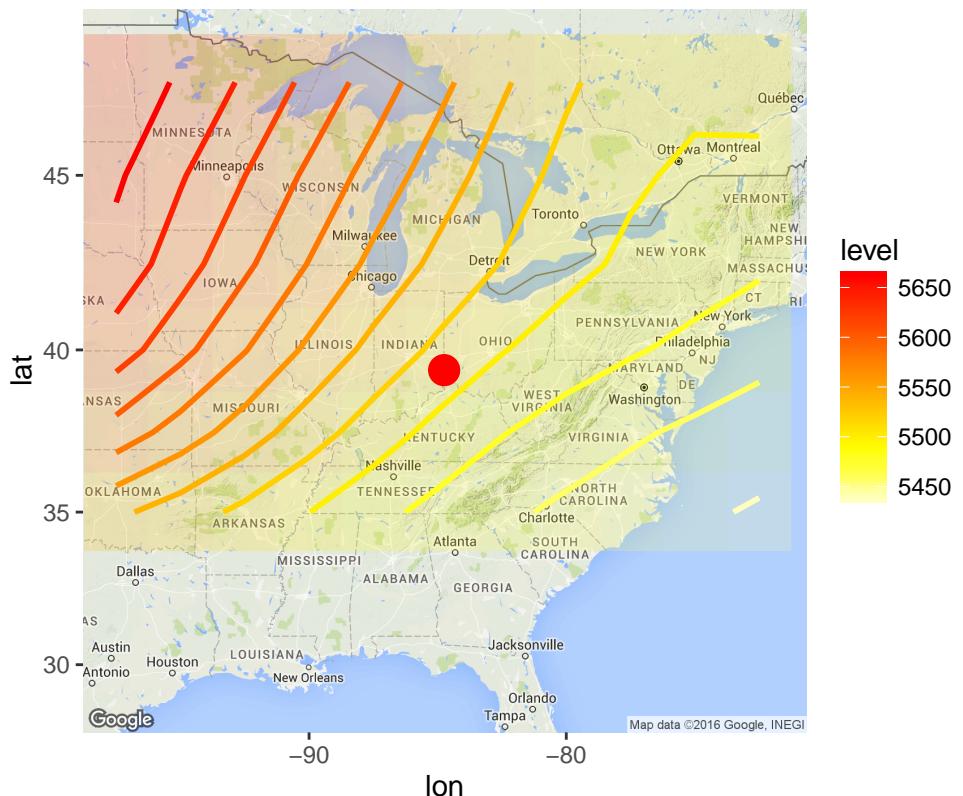
```
draw_map(target, "usa", tmp_xlon, tmp_ylat, tmp_time, 1, '5 days before the flood happened')
```

5 days before the flood happened



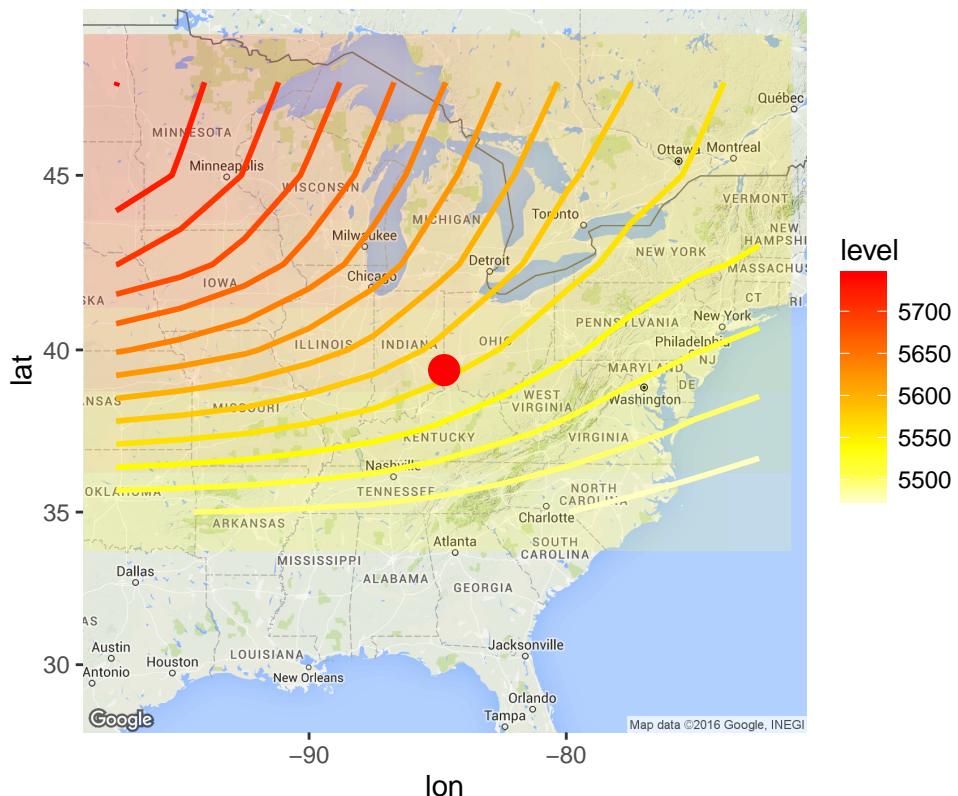
```
draw_map(target, "usa", tmp_xlon, tmp_ylat, tmp_time, 6, 'First day when the flood happened')
```

First day when the flood happened



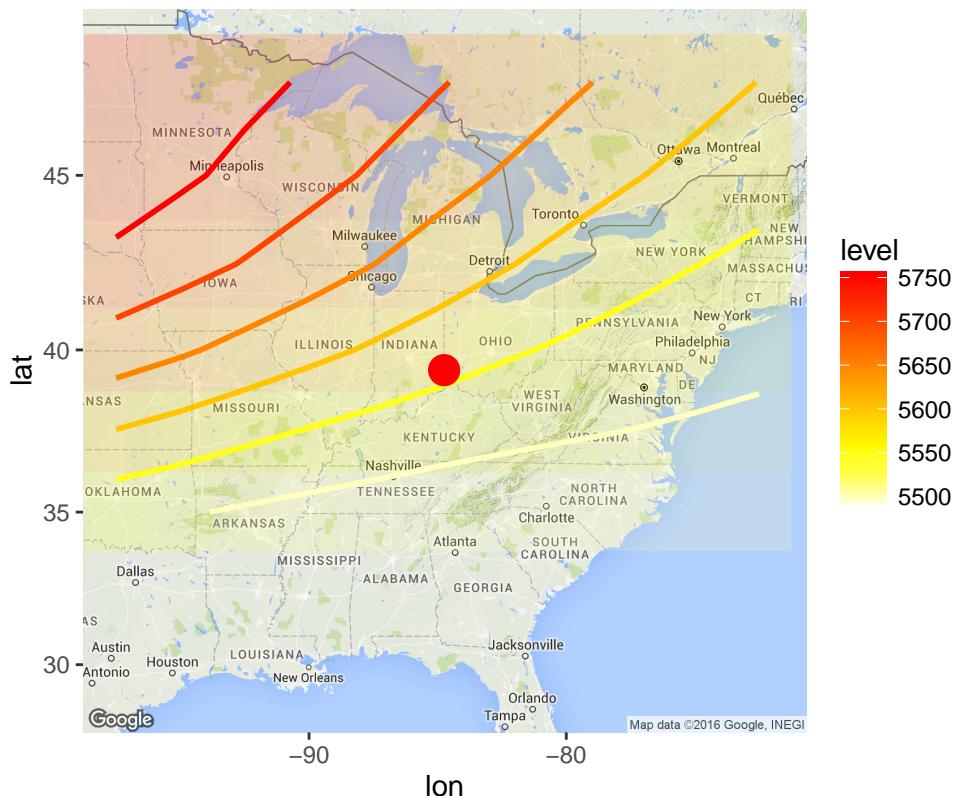
```
draw_map(target, "usa", tmp_xlon, tmp_ylat, tmp_time, 7, 'Second day when the flood happened')
```

Second day when the flood happened



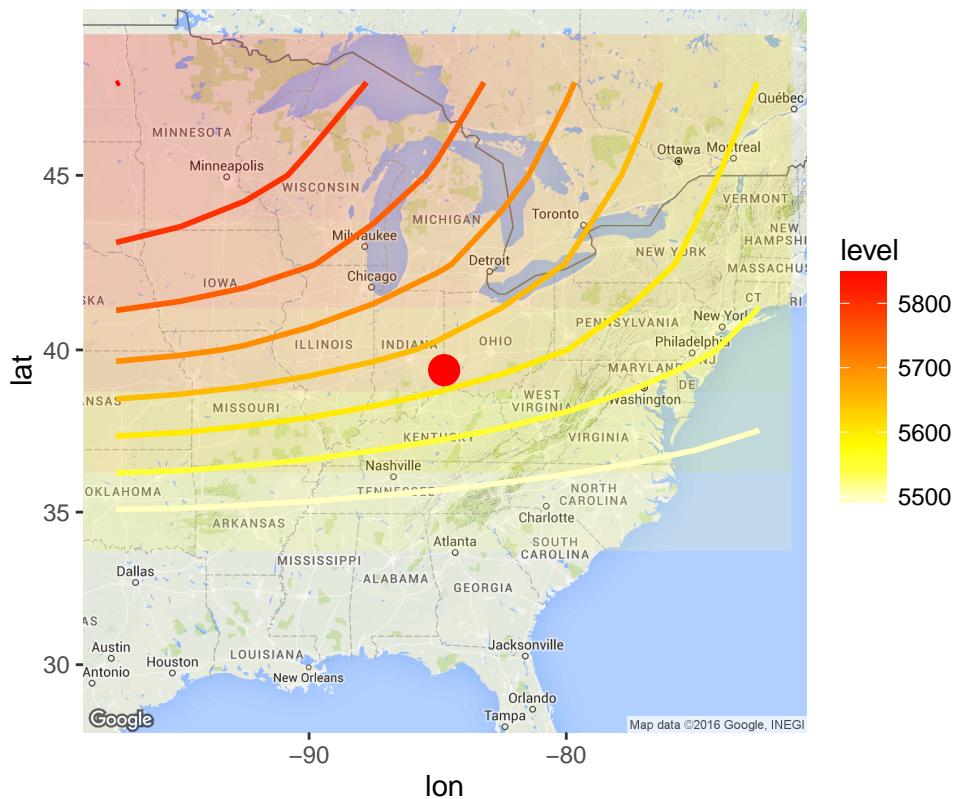
```
draw_map(target, "usa", tmp_xlon, tmp_ylat, tmp_time, 8, 'Third day when the flood happened')
```

Third day when the flood happened



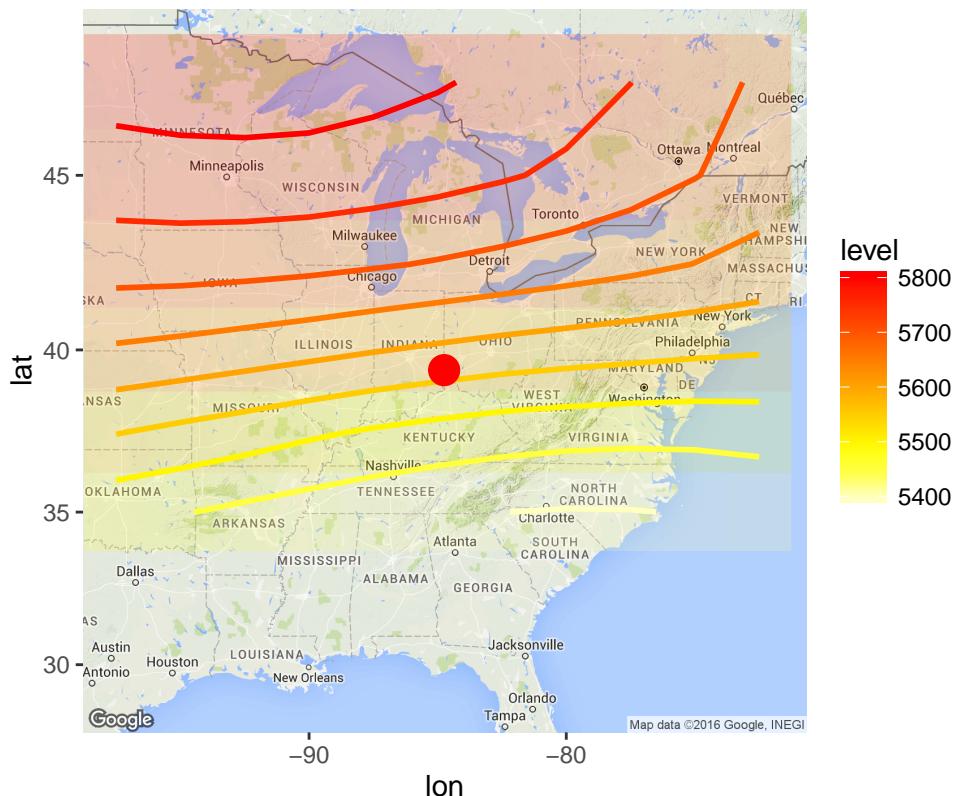
```
draw_map(target, "usa", tmp_xlon, tmp_ylat, tmp_time, 9, 'One day after the flood happened')
```

One day after the flood happened



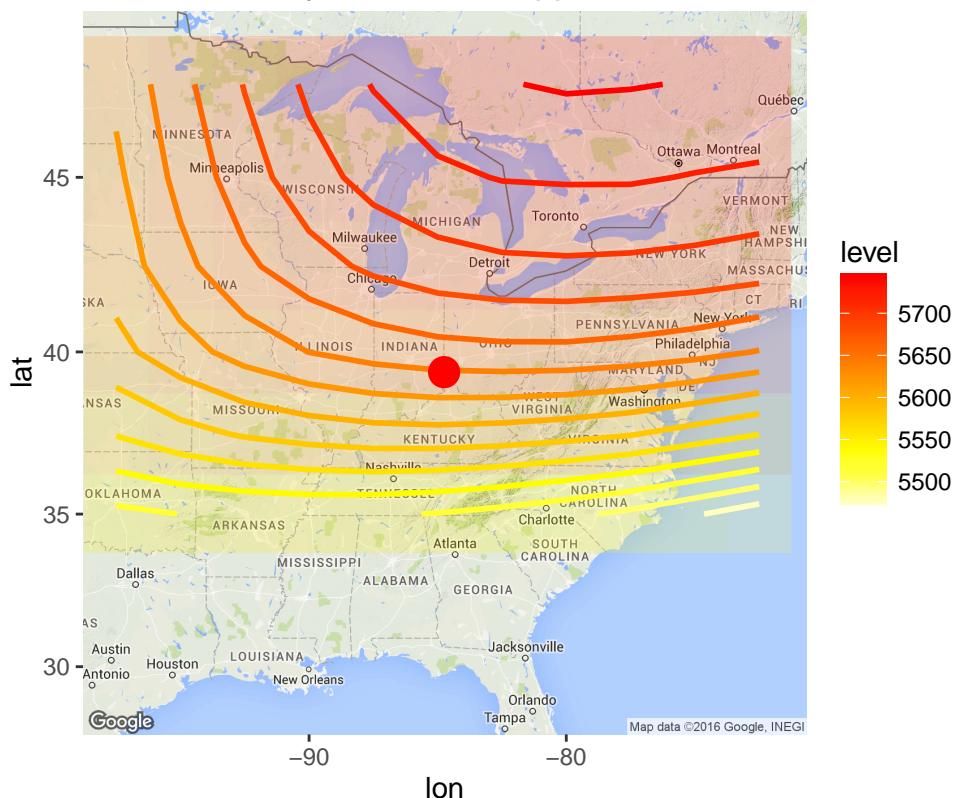
```
draw_map(target, "usa", tmp_xlon, tmp_ylat, tmp_time, 10, 'Two days day when flood happened')
```

Two days day when flood happened



```
draw_map(target, "usa", tmp_xlon, tmp_ylat, tmp_time, 1 + flood_end + 5 - (flood_begin - 5), '5 day after'
```

5 day after flood happened



As seen above, we can see changes of contour when the flood happened. At the second day of the flood, the density of contour is high.

TODO 5, PCA

Jordan